

NIDS/NIPS and Web Proxy Analysis
University of Maryland Baltimore County

Presented to: (professor's info removed)

April 17, 2020

Table of Content

1. Introduction
2. Analysis
3. Conclusion
4. Glossary
5. Reference

Introduction

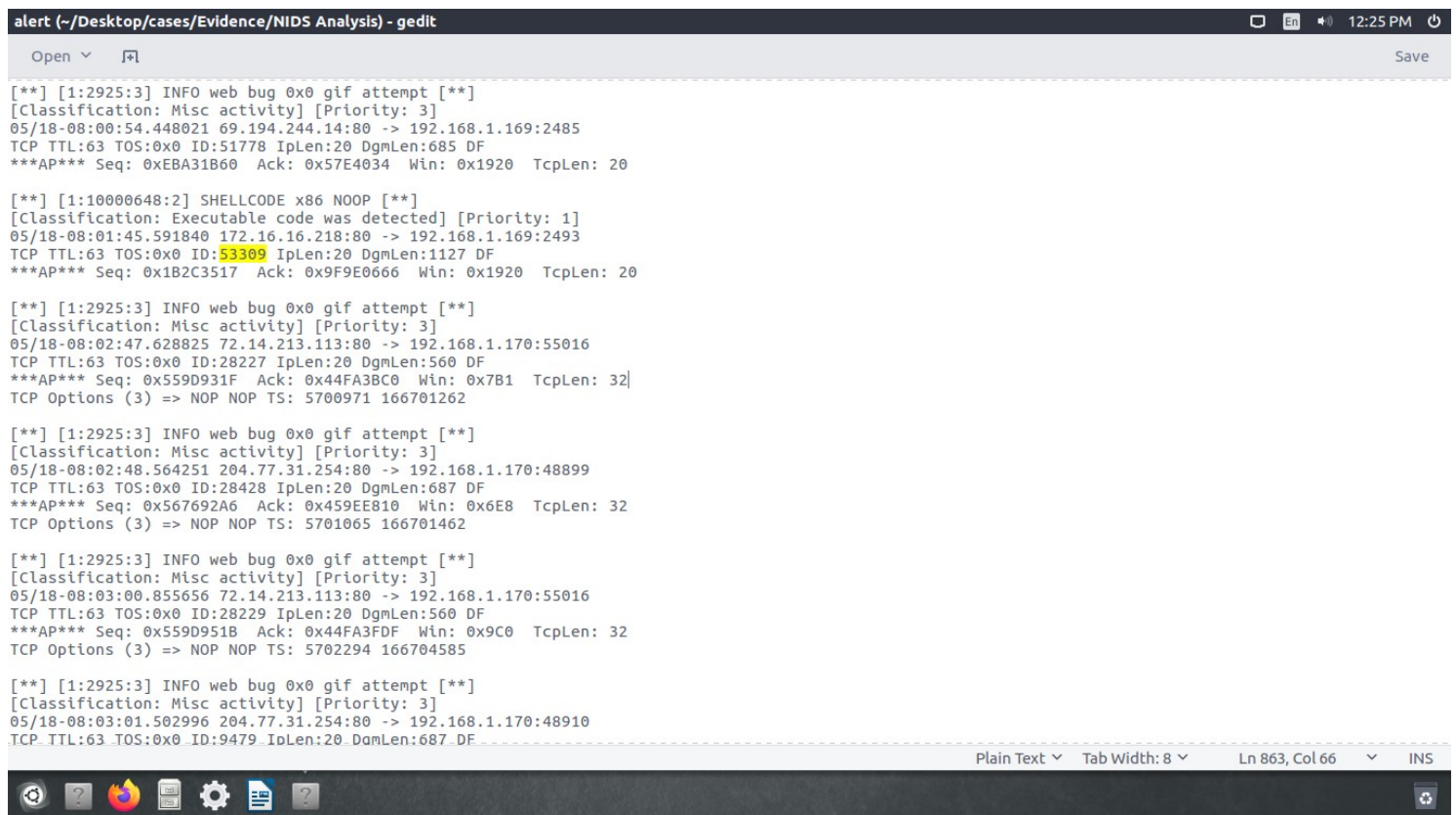
While working with Division of IT at University of Maryland Baltimore County, for their Digital Forensics and Incident Response team, I have been tasked to a) Perform log analysis on IDS/IPS alerts (Snort) to become familiar with alert formats and using them to investigate potentially malicious traffic and b) Perform log analysis on Proxy logs (Squid) to become familiar with proxy log formats, and how to review them to investigate web related traffic. The methodology adopted for the analysis of the given alert, tcpdump.log, snort.conf, access.log and store.log files is done by using a combination of tools and linux terminal commands– wireshark, hex editor, grep command and MS Excel. This task encompasses in detail analysis of the logs and alerts generated by Network Intrusion Protection and Detection Systems (NIDS/NIPS).

Analysis

1. IDS/IPS Alert Background (Part 1)

1.1 Examine the alert's data to understand the logistical context

Figure 1.1.1, we analyse the alert and note: Source IP: 172.16.16.218 (Source Port :80) and Destination IP: 192.168.1.169:2493, Packet Delivery ID : 53309, Snort ID: 10000648, IPlink : 20 and Datagram link : 1127



```
alert (~/.Desktop/cases/Evidence/NIDS Analysis) - gedit
Open  Save

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:00:54.448021 69.194.244.14:80 -> 192.168.1.169:2485
TCP TTL:63 TOS:0x0 ID:51778 IpLen:20 DgmLen:685 DF
***AP*** Seq: 0xEBA31B60 Ack: 0x57E4034 Win: 0x1920 TcpLen: 20

[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 172.16.16.218:80 -> 192.168.1.169:2493
TCP TTL:63 TOS:0x0 ID:53309 IpLen:20 DgmLen:1127 DF
***AP*** Seq: 0x1B2C3517 Ack: 0x9F9E0666 Win: 0x1920 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:02:47.628825 72.14.213.113:80 -> 192.168.1.170:55016
TCP TTL:63 TOS:0x0 ID:28227 IpLen:20 DgmLen:560 DF
***AP*** Seq: 0x559D931F Ack: 0x44FA3BC0 Win: 0x7B1 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5700971 166701262

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:02:48.564251 204.77.31.254:80 -> 192.168.1.170:48899
TCP TTL:63 TOS:0x0 ID:28428 IpLen:20 DgmLen:687 DF
***AP*** Seq: 0x567692A6 Ack: 0x459EE810 Win: 0x6E8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5701065 166701462

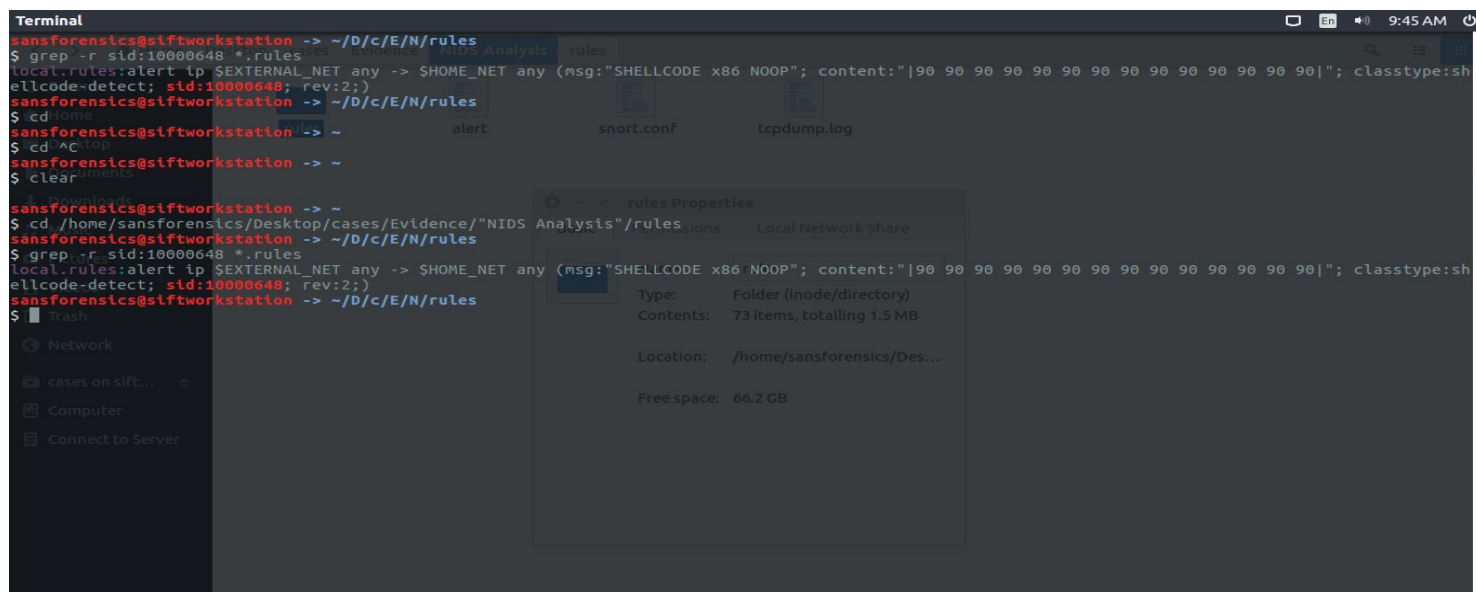
[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:03:00.855656 72.14.213.113:80 -> 192.168.1.170:55016
TCP TTL:63 TOS:0x0 ID:28229 IpLen:20 DgmLen:560 DF
***AP*** Seq: 0x559D951B Ack: 0x44FA3FDF Win: 0x9C0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5702294 166704585

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:03:01.502996 204.77.31.254:80 -> 192.168.1.170:48910
TCP TTL:63 TOS:0x0 ID:9479 IpLen:20 DgmLen:687 DF
```

Figure 1.1.1

1.2 Compare the alert to the rule, in order to better understand WHAT it has been built to detect:

Figure 1.2.1, using the snort ID we use `cd /home/sansforensics/Desktop/cases/Evidence/"NIDS Analysis"/rules` and then `grep -r sid:10000648 *.rules` to find the `shellcode-detect` among a list of `.rules` files and understand more about alert and the rule using the sid and find that `local.rule` file mentions the `classtype : shellcode-detect`



```
Terminal
sansforensics@siftworkstation -> ~/D/c/E/N/rules
$ grep -r sid:10000648 *.rules
local.rules:alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; classtype:shellcode-detect; sid:10000648; rev:2;)
sansforensics@siftworkstation -> ~/D/c/E/N/rules
$ cd /home
sansforensics@siftworkstation -> ~
$ cd /home/Desktop
sansforensics@siftworkstation -> ~
$ clear

sansforensics@siftworkstation -> ~
$ cd /home/sansforensics/Desktop/cases/Evidence/"NIDS Analysis"/rules
sansforensics@siftworkstation -> ~/D/c/E/N/rules
$ grep -r sid:10000648 *.rules
local.rules:alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; classtype:shellcode-detect; sid:10000648; rev:2;)
sansforensics@siftworkstation -> ~/D/c/E/N/rules
$
```

Figure 1.2.1

1.3 Retrieve the packet that triggered the alert

Figure 1.3.1 and 1.3.2, we open the `tcpdump.log` file in Wireshark and use the filter `ip.id==53309` to analyse the packet and we were able to find `Seq: 0x1B2C3517` and `Ack: 0x9F9E0666` within TCP hex dump.

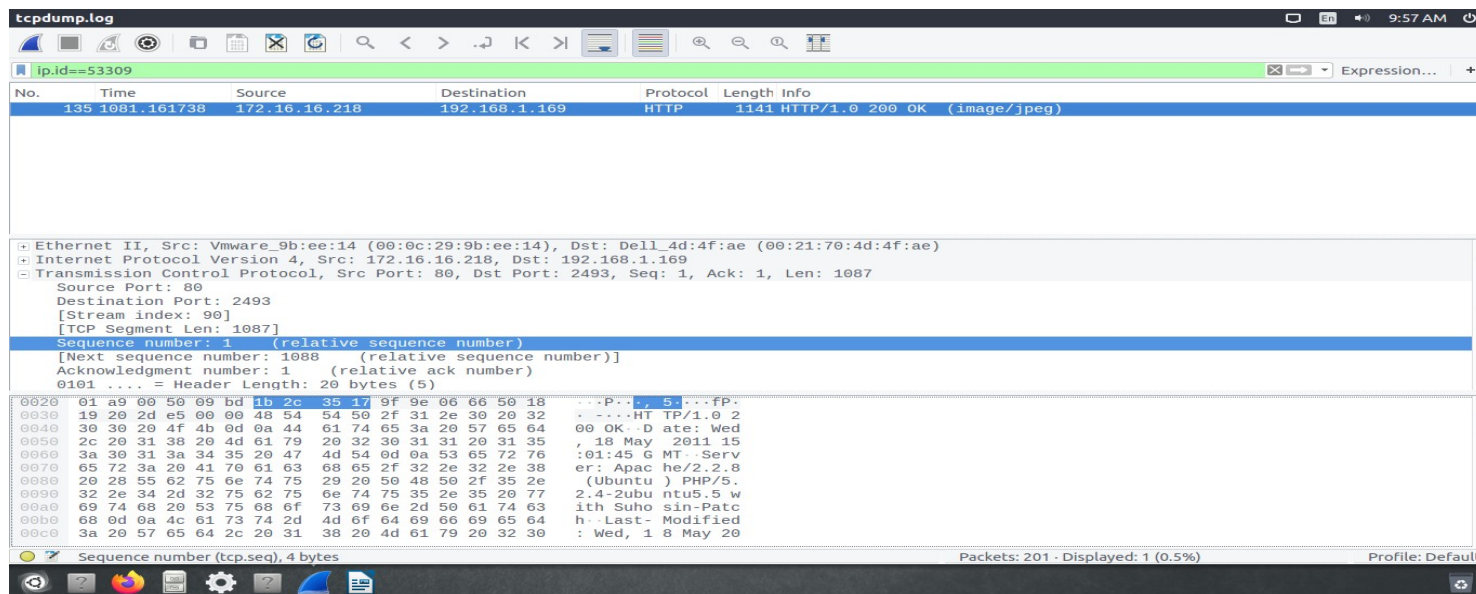


Figure 1.3.1

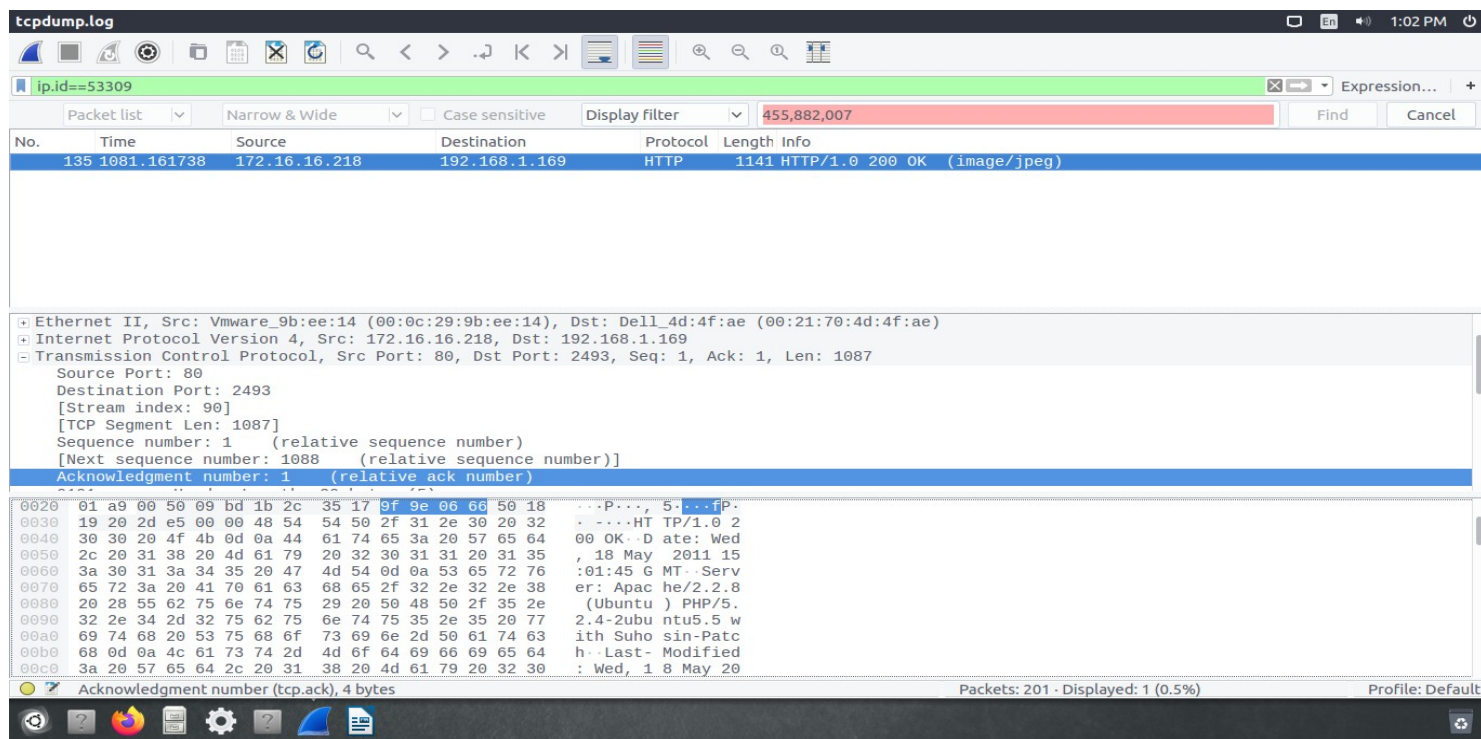


Figure 1.3.2

1.4 Compare the rule to the packet to understand WHY it fired

Figure 1.4.1 and 1.4.2, we compare the rule and packet and find that the rule is set to detect suspicious content from an external IP to port 80 and here an image file is being transferred with potentially malicious shellcode.

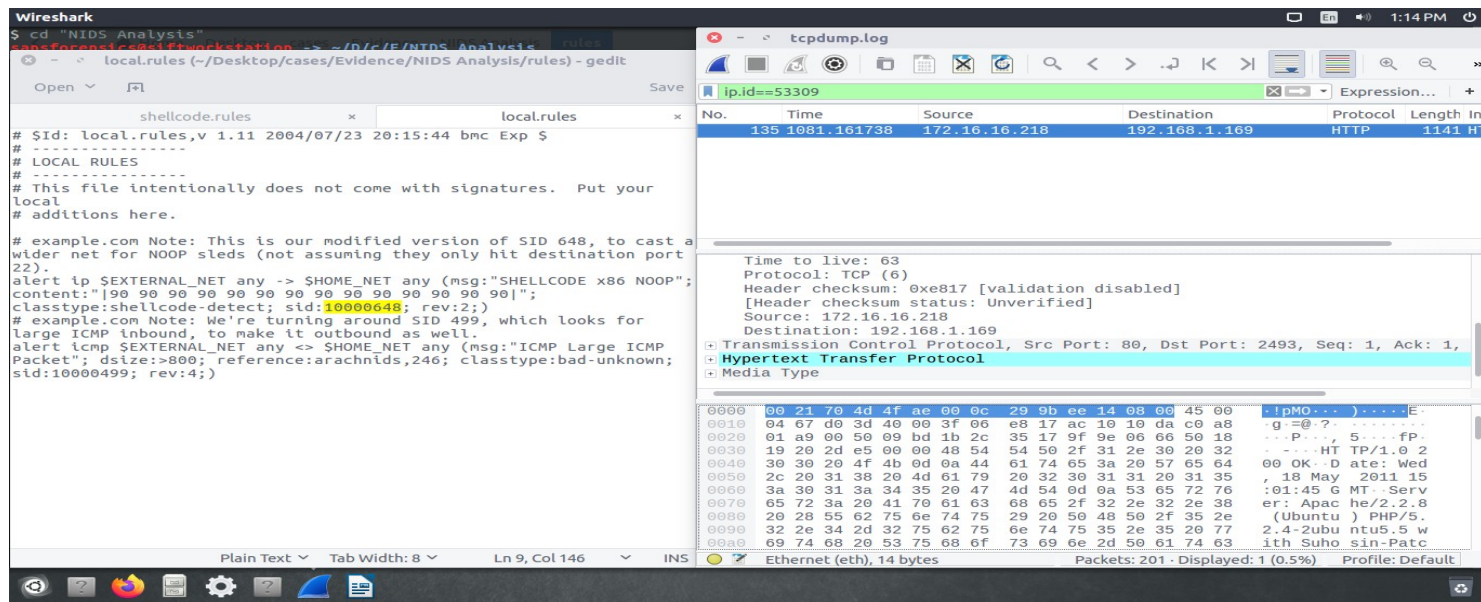


Figure 1.4.1

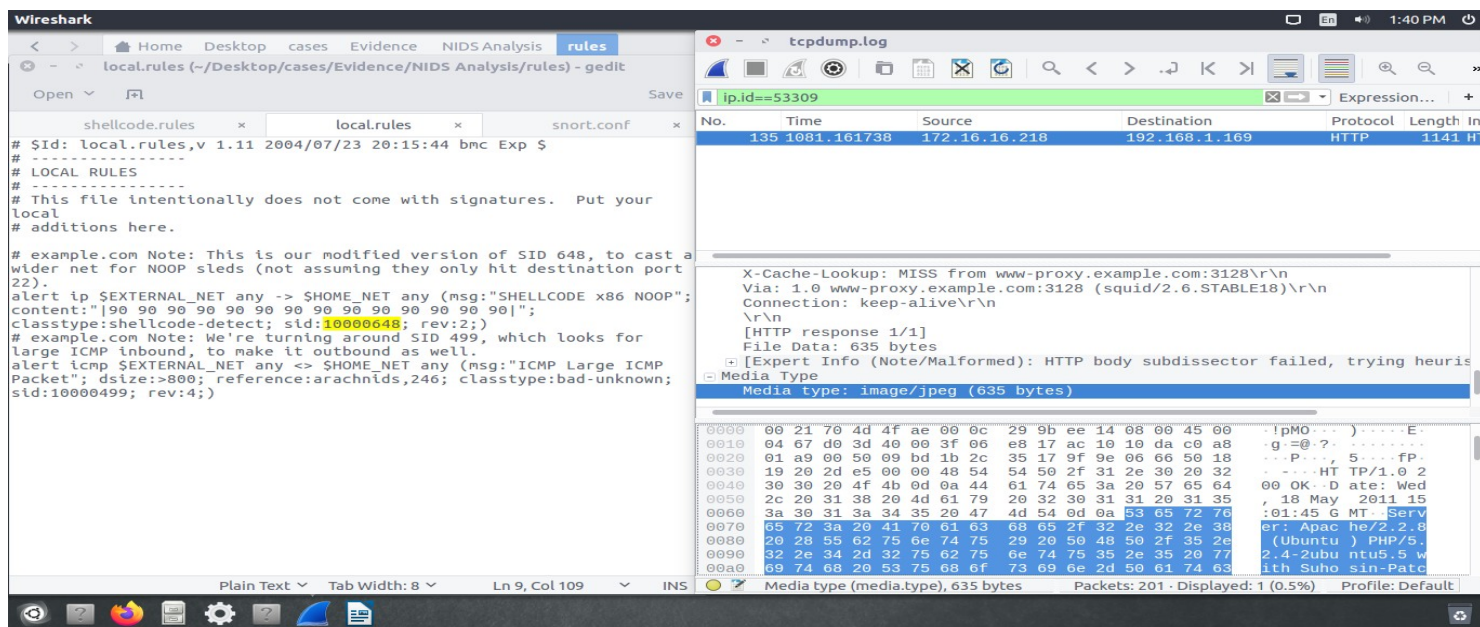


Figure 1.4.2

1.5 Construct a timeline of alerted activities involving the potentially malicious outside host

In Figure 1.5.1, 1.5.2, 1.5.3 and 1.5.4:

- at 07:43:51 we detect “COMMUNITY MISC BAD-SSL tcp detect [**] [Classification: Misc activity]” from 204.11.50.137:80 to internal host 192.168.1.169
- From 07:43:52 to 08:15:08, we detect “INFO web bug 0x0 gif attempt [**] [Classification: Misc activity]” from different external hosts to internal hosts like 192.168.1.169, 192.168.1.170etc
- In between the above time frame, at 08:01:45 MST we also detect “SHELLCODE x86 NOOP [**] [Classification: Executable code was detected]”, from an external web server 172.16.16.218:80 to 192.168.1.169

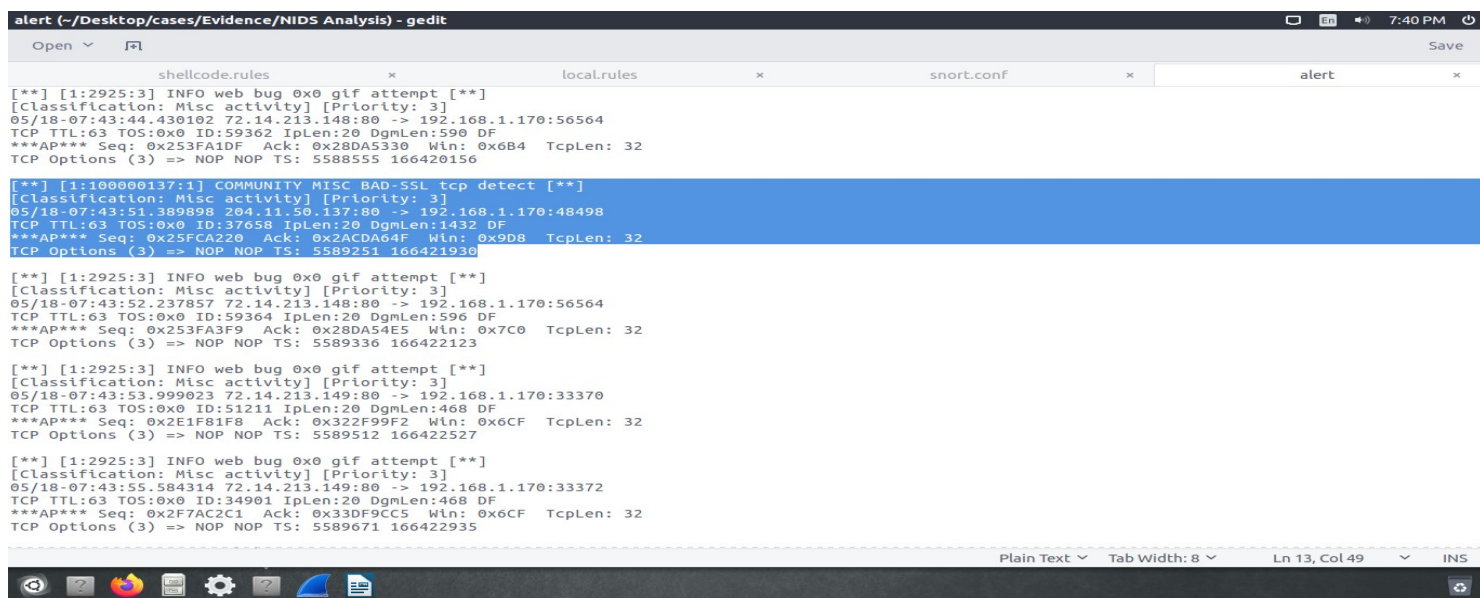
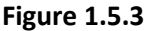
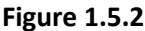


Figure 1.5.1




```
alert (~/Desktop/cases/Evidence/NIDS Analysis) - gedit
Open  Save
shellcode.rules  local.rules  snort.conf  alert
****AP*** Seq: 0xBE039E0D Ack: 0xABF0BCC8 Win: 0x7DA TcpLen: 32
TCP Options (3) => NOP NOP TS: 5739479 166797574

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:09:12.925492 72.14.213.113:80 -> 192.168.1.170:58142
TCP TTL:63 TOS:0x0 ID:3526 Iplen:20 Dgmlen:560 DF
****AP*** Seq: 0xBE03A009 Ack: 0xABFDC52F Win: 0xD82 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5739499 166797607

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:09:12.985289 64.94.107.27:80 -> 192.168.1.170:52865
TCP TTL:63 TOS:0x0 ID:1496 Iplen:20 Dgmlen:498 DF
****AP*** Seq: 0xBDD4960A Ack: 0xAC31A280 Win: 0x776 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5739505 166797609

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:15:06.474654 64.30.224.42:80 -> 192.168.1.169:2634
TCP TTL:63 TOS:0x0 ID:24543 Iplen:20 Dgmlen:639 DF
****AP*** Seq: 0x5EA4839 Ack: 0x2CDA0DE Win: 0x2180 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:15:08.286168 216.239.113.95:80 -> 192.168.1.169:2650
TCP TTL:63 TOS:0x0 ID:57018 Iplen:20 Dgmlen:728 DF
****AP*** Seq: 0x95FC010 Ack: 0x9C6308FA Win: 0x1A28 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-08:15:08.361442 138.108.28.10:80 -> 192.168.1.169:2649
TCP TTL:63 TOS:0x0 ID:46682 Iplen:20 Dgmlen:545 DF
****AP*** Seq: 0xA06A8C7 Ack: 0xB9C8BDC5 Win: 0x1B28 TcpLen: 20

Plain Text  Tab Width: 8  Ln 1295, Col 66  INS
```

Figure 1.5.4

1.6 Construct a timeline of alerted activities involving the target

Figure 1.6.1, 1.6.2:

- at 08:01:45 MST we detect “SHELLCODE x86 NOOP [**] [Classification: Executable code was detected]”, from an external web server 172.16.16.218:80 to 192.168.1.169. This signifies the malicious intent to inject shellcode.
- and at 08:15:08, we detect “INFO web bug 0x0 gif attempt [**] [Classification: Misc activity]” from different external hosts to internal host 192.168.1.169. This type of alert was also found at many other instances.

```
alert (~/Desktop/cases/Evidence/NIDS Analysis) - gedit
Open  Save
shellcode.rules  local.rules  snort.conf  alert
[Classification: Misc activity] [Priority: 3]
05/18-07:45:00.179227 207.171.185.201:80 -> 192.168.1.170:59891
TCP TTL:63 TOS:0x0 ID:9883 Iplen:20 Dgmlen:693 DF
****AP*** Seq: 0x6AFC454F Ack: 0x711BD654 Win: 0x6B4 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5596130 166439099

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:45:09.351488 207.46.140.21:80 -> 192.168.1.169:2127
TCP TTL:63 TOS:0x0 ID:5298 Iplen:20 Dgmlen:607 DF
****AP*** Seq: 0x72F77253 Ack: 0xF2E55562 Win: 0x1B20 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:45:09.834604 207.46.193.178:80 -> 192.168.1.169:2132
TCP TTL:63 TOS:0x0 ID:65214 Iplen:20 Dgmlen:455 DF
****AP*** Seq: 0x739099DC Ack: 0x1D01478F Win: 0x1920 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:45:10.109334 204.236.235.239:80 -> 192.168.1.170:50817
TCP TTL:63 TOS:0x0 ID:55589 Iplen:20 Dgmlen:442 DF
****AP*** Seq: 0x53F38488 Ack: 0x5A3E2668 Win: 0x7D2 TcpLen: 32
TCP Options (3) => NOP NOP TS: 5597123 166439220

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:45:51.204013 65.54.95.43:80 -> 192.168.1.169:2142
TCP TTL:63 TOS:0x0 ID:48309 Iplen:20 Dgmlen:809 DF
****AP*** Seq: 0x924FC8F4 Ack: 0x24D9F700 Win: 0x1920 TcpLen: 20

[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:46:15.877801 207.46.140.21:80 -> 192.168.1.169:2151
TCP TTL:63 TOS:0x0 ID:12736 Iplen:20 Dgmlen:607 DF

Plain Text  Tab Width: 8  Ln 117, Col 67  INS
```

Figure 1.6.1

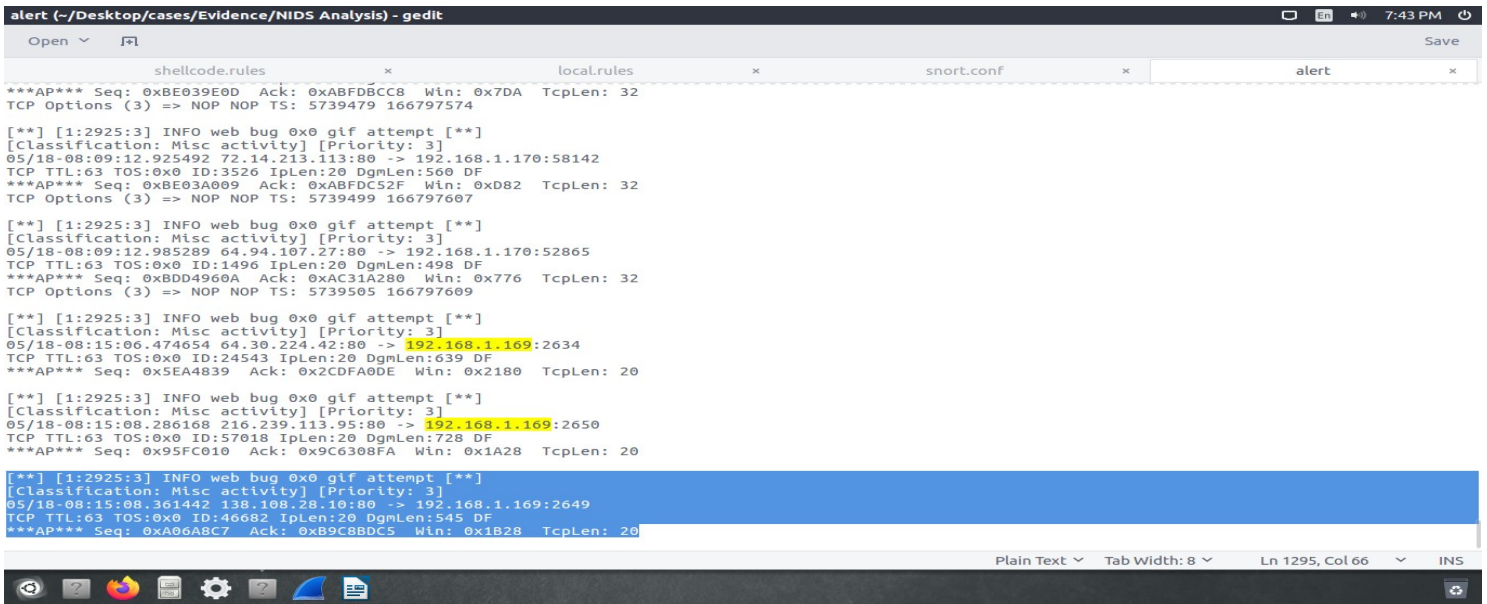


Figure 1.6.2

2. Proxy Log Background (Part 2)

2.1 Determine whether the evidence extracted from the Squid cache corroborates our findings from the Snort logs.

Figure 2.1.1, an unusual binary sequence that is commonly associated with buffer overflow exploits was found in ETag in the external webserver's HTTP response : **1238-27b-4a38236f5d880**.

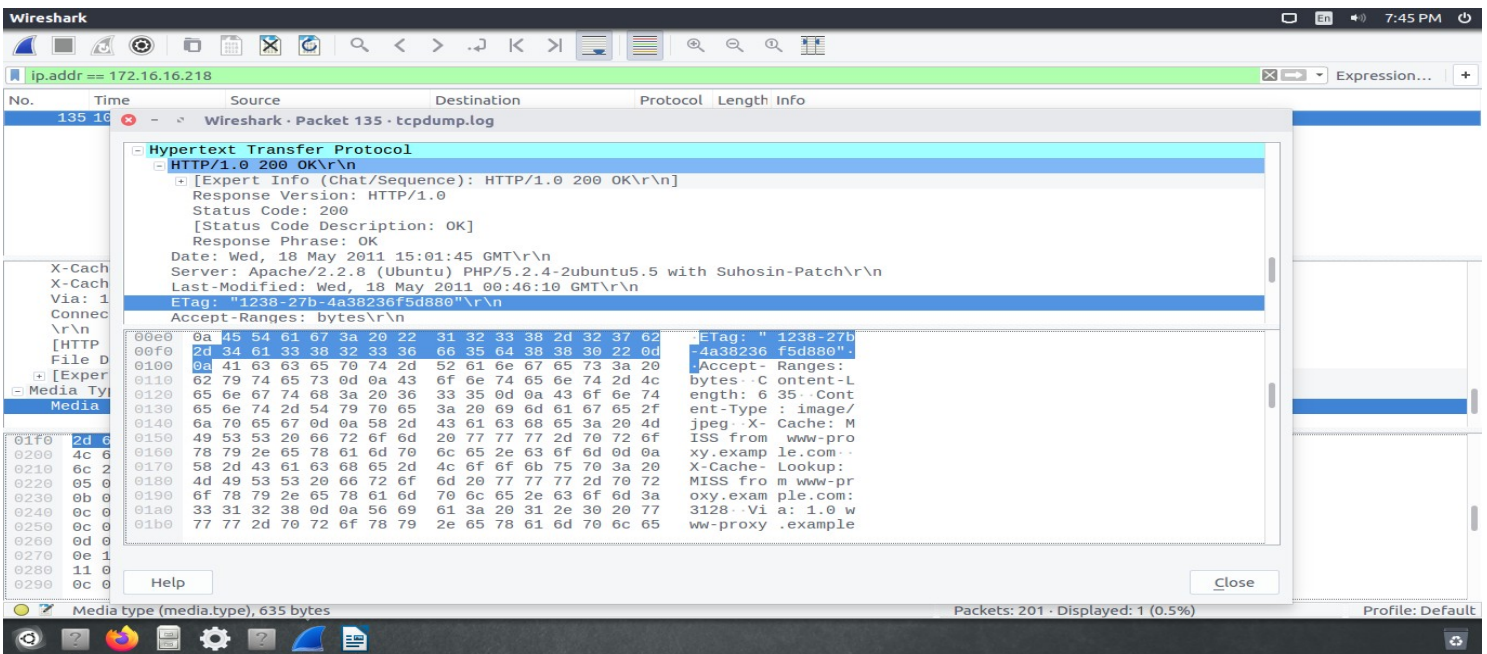


Figure 2.1.1

Figure 2.1.2, we use `cd /home/sansforensics/Desktop/cases/Evidence/"Proxy Log Analysis"/squid` and then `grep -r '1238-27b-4a38236f5d880'` to look for file with the matching ETag sequence and we find that a binary file with location "00/05/0000058A" has matched our search.

```

Terminal
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-transactions4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-enterpriseservices4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-numeric4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-data4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-runtime-serialization-formatters-soap4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-webbrowser4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/n/mono/libmono-system-windows-forms4.0-cil_4.2.1.102+dfsg2-7ubuntu4_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
sansforensics@siftworkstation:~$ cd /home/sansforensics/Desktop/cases/Evidence/Proxy Log Analysis/squid
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r '1238-27b-4a38236f5d880'
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r '1238-27b-4a38236f5d880'
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r '1238-27b-4a38236f5d880'
binary file 00/05/0000058A matches
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$

```

Figure 2.1.2

Figure 2.1.3, we open the binary file 0000058A, using Bless Hex Editor and could verify the matching ETag sequence - 1238-27b4a38236f5d880. We were also able to find the location `http://www.evil.evl/pwnj.jpg` of the malicious JPEG file, pwnj.jpg in ASCII characters.

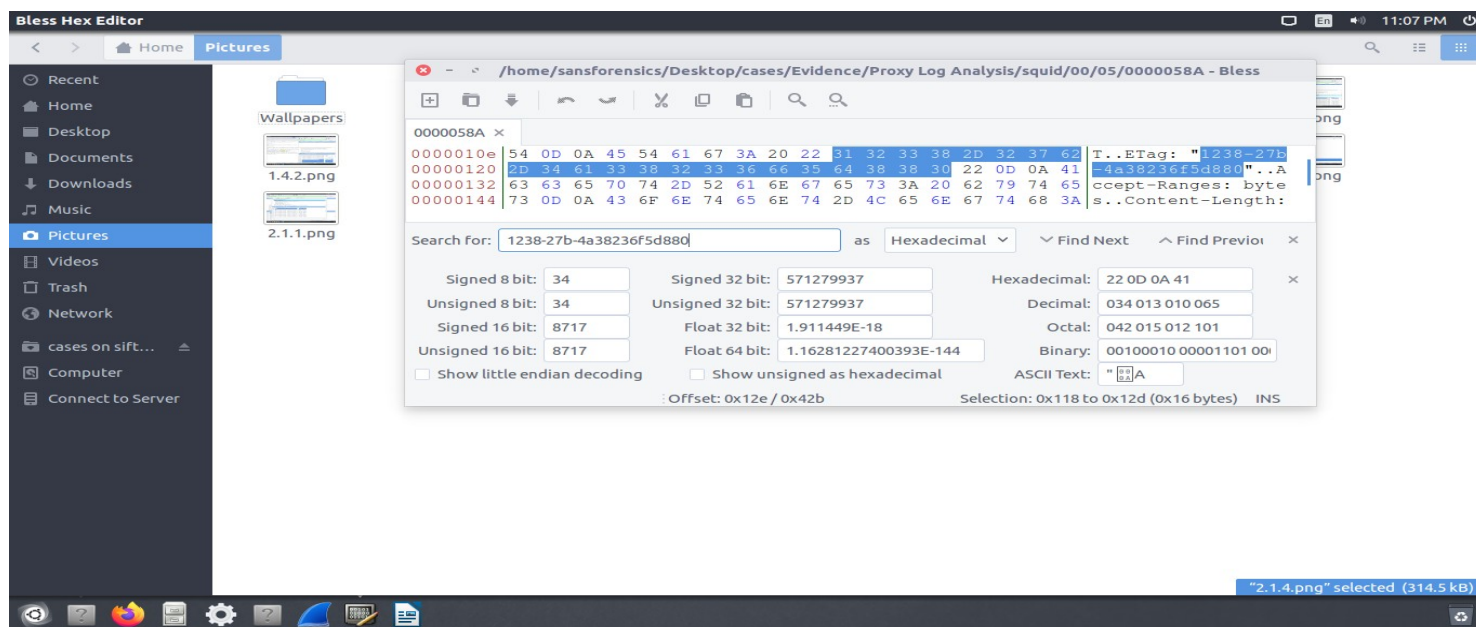


Figure 2.1.3

Figure 2.1.4, using `cd /home/sansforensics/Desktop/cases/Evidence/"Proxy Log Analysis"/squid` and then `grep -r "http://www.evil.evl/pwny.jpg"` we search for the .jpg file's url among other cache files in the squid directory and find two binary file matches **0000058A** and **00000589**.

```

sansforensics@siftworkstation ~
$ cd /home/sansforensics/Desktop/cases/Evidence/"Proxy Log Analysis"/squid
sansforensics@siftworkstation ~/D/c/Evidence
$ cd Proxy Log Analysis
bash: cd: Proxy: No such file or directory
sansforensics@siftworkstation ~/D/c/Evidence
$ cd "Proxy Log Analysis"
sansforensics@siftworkstation ~/D/c/E/Proxy Log Analysis
$ cd squid
sansforensics@siftworkstation ~/D/c/E/P/squid
$ grep -r '1238-27b4a38236f5d880'
sansforensics@siftworkstation ~/D/c/E/P/squid
$ grep -r '1238-27b4a38236f5d880'
sansforensics@siftworkstation ~/D/c/E/P/squid
$ grep -r '1238-27b-4a38236f5d880'
Binary file 00/05/0000058A matches
sansforensics@siftworkstation ~/D/c/E/P/squid
$ grep -r "http://www.evil.evl/pwny.jpg"
Binary file 00/05/0000058A matches
Binary file 00/05/00000589 matches
sansforensics@siftworkstation ~/D/c/E/P/squid
$
  
```

Figure 2.1.4

2.2 Present any information you can find regarding the identity of any internal users who have been engaged in suspicious activities.

Figure 2.2.1 and 2.2.2, we copy the **access.log** and **store.log** files from **var-log-squid.zip** to a Windows machine to open them using MS Excel and filter them using 192.168.1.169. Then we hit and try different epoch timestamps to match with the alert timestamp. For this we use an online converter (**figure 2.2.3**) and find out that **1305729906**, **1305730906** and **1305731047** (blue color coded) are the closest timestamps corresponding to **07:45:06** (when internal host 192.168.1.169 started browsing external web sites), **08:01:46** (when alert triggered for shellcode injection) and **08:04:07** (when internal host 192.168.1.169 sent crafted packets to other internal hosts).

Line	Timestamp	IP	Request
1	1305729905	192.168.1.1	TCP_MISS/2 3521 GET http://cool.stb.s-mn.com/146/68063C0AED19CD76AF08F335045CA.jpg - DIRECT/165.54.95.154 image/jpeg
2	1305729906	192.168.1.1	TCP_MISS/2 5975 GET http://www.bing.com/1as/895538f6en.js - DIRECT/1704.203.18.163 application/javascript
3	1305729906	192.168.1.1	TCP_MISS/2 4640 GET http://cool.stb.s-mn.com/146/68063C0AED19CD76AF08F335045CA.jpg - DIRECT/165.54.95.154 image/jpeg
4	1305729906	192.168.1.1	TCP_MISS/2 5975 GET http://www.bing.com/1as/895538f6en.js - DIRECT/1704.203.18.163 application/javascript
5	1305729926	192.168.1.1	TCP_MISS/2 1447 GET http://www.msn.com/ADSAdClient31.dll? - DIRECT/165.55.17.25 text/html
6	1305729931	192.168.1.1	TCP_MISS/2 3389 GET http://rad.msn.com/ADSAdClient31.dll? - DIRECT/165.55.5.233 text/html
7	1305729934	192.168.1.1	TCP_MISS/2 1916 GET http://rad.msn.com/ADSAdClient31.dll? - DIRECT/165.55.5.232 text/html
8	1305730890	192.168.1.1	TCP_MISS/2 9535 GET http://sketchy.evl/ - DIRECT/172.16.16.217 text/html
9	1305730890	192.168.1.1	TCP_MISS/2 1926 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
10	1305730890	192.168.1.1	TCP_MISS/2 1095 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
11	1305730891	192.168.1.1	TCP_MISS/2 4412 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
12	1305730906	192.168.1.1	TCP_MISS/2 2005 GET http://www.bluehost.com/servlet/GenerateContentCodes?CDS=125&CDS2=1 - NONE/- text/javascript
13	1305730906	192.168.1.1	TCP_MISS/2 6107 GET http://www.cdn.cloudflare.com/ - DIRECT/172.16.16.217 text/html
14	1305730906	192.168.1.1	TCP_MISS/2 620 GET http://sketchy.evl/wp-content/images/milestones/cvark.gif - DIRECT/172.16.16.217 image/gif
15	1305730906	192.168.1.1	TCP_MISS/2 3474 GET http://pixel.quantserve.com/pixel.gif?1338838344;ip=192.168.1.1;pa=PO-1285330934-1305730907705;na=1;url=http://3A%2F%2Fbdv.bidvertiser.com%2Fbidvertiser.dbm%3Fpid%3D67754%26bid%3D272692%26RD%3D939 - DIRECT/172.16.16.217 image/gif
16	1305730945	192.168.1.1	TCP_MISS/2 2480 GET http://sketchy.evl/wp-login.php - DIRECT/172.16.16.217 text/html
17	1305730945	192.168.1.1	TCP_MISS/2 20247 GET http://sketchy.evl/wp-admin/vp-admin.css? - DIRECT/172.16.16.217 text/css
18	1305730945	192.168.1.1	TCP_MISS/2 19733 GET http://sketchy.evl/wp-admin/images/login-bg.gif - DIRECT/172.16.16.217 image/gif
19	1305731046	192.168.1.1	TCP_MISS/2 331 GET http://ad.z5x.net/imp? - DIRECT/166.94.245.1 -
20	1305731046	192.168.1.1	TCP_MISS/2 306 GET http://ak1.abn.net/icontent/yieldmanager.com/ak1.gif? - DIRECT/204.203.18.155 image/gif
21	1305731046	192.168.1.1	TCP_MISS/2 653 GET http://icontent.yieldmanager.com/ak1.gif? - DIRECT/204.203.18.155 image/gif
22	1305731047	192.168.1.1	TCP_MISS/2 2242 GET http://ad.yieldmanager.com/imp? - DIRECT/178.137.51.1 application/javascript
23	1305731047	192.168.1.1	TCP_MISS/2 5036 GET http://icontent.yieldmanager.edgesuite.net/atom/51b326b579b326b5230a5c6d89fe7cd0dd04260e.gif - DIRECT/204.203.18.155 image/gif
24	1305731108	192.168.1.1	TCP_MISS/2 608 POST http://sketchy.evl/wp-comments-post.php - DIRECT/172.16.16.217 text/html
25	1305731108	192.168.1.1	TCP_MISS/2 9576 GET http://sketchy.evl/ - DIRECT/172.16.16.217 text/html
26	1305731108	192.168.1.1	TCP_MISS/2 1926 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
27	1305731108	192.168.1.1	TCP_MISS/2 1926 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
28	1305731108	192.168.1.1	TCP_MISS/2 1113 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
29	1305729793	192.168.1.1	TCP_MISS/2 3469 GET http://start.ubuntu.com/8.04/ - DIRECT/91.189.90.41 text/html
30	1305729793	192.168.1.1	TCP_MISS/2 1635 GET http://start.ubuntu.com/8.04/style.css - DIRECT/91.189.90.41 text/css
31	1305729800	192.168.1.1	TCP_MISS/2 1926 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/172.16.200.193.154 text/html
32	1305729800	192.168.1.1	TCP_MISS/2 3052 GET http://start.ubuntu.com/8.04/screen-Bc-24-24-12.css? - DIRECT/91.189.90.41 text/css
33	1305729800	192.168.1.1	TCP_MISS/2 887 GET http://start.ubuntu.com/8.04/images/ba-paoc.png - DIRECT/91.189.90.41 image/png

Figure 2.2.1

After careful analysis of activities from both access.log and store.log files around the aforementioned timestamps, we find out that the internal user was browsing images on various external websites, among which also happened to show some activity with .evl websites like <http://sketchy.evl/?>, www.evil.evl/pwny.jpg (.evl is a top level domain used by Evil Systems). This user also seems to have admin access to one of the .evl website because there are traces of login attempts to admin profiles.

Figure 2.2.2

time zones to calculate timestamps. Most programming languages have libraries to help you converting time zones, calculating by hand might not be a good idea because of the variety of time zones en daylight saving times. But here's a list of time zones and offset in seconds.

Convert epoch to other time zone

Convert to time zone

Conversion results (1305730906)

1305730906 converts to **Wednesday May 18, 2011 08:01:46 (am)** in time zone **America/Los Angeles (PDT)**
The offset (difference to Greenwich Time/GMT) is -07:00 or in seconds -25200.
This date is in daylight saving time.

Other time zones

Pages

- Home
- Preferences
- Toggle theme

Tools

- Epoch converter
- Batch converter
- Time zone converter
- Timestamp list
- LDAP converter
- Webkit/Chrome timestamp
- Unix hex timestamp
- Cocoa Core Data timestamp
- Mac HFS+ timestamp
- SAS timestamp
- Seconds/days since year 0
- Bin/Oct/Hex converter
- Countdown in seconds
- Epoch clock

Date and Time

- Week numbers
- Weeks by year
- Day numbers
- Days by year

Figure 2.2.3

Conclusion

The NIDS/NIPS and Web Proxy Analysis lab taught me what tools I can use to to to a) Perform log analysis on IDS/IPS alerts (Snort) to become familiar with alert formats and using them to investigate potentially malicious traffic and b) Perform log analysis on Proxy logs (Squid) to become familiar with proxy log formats, and how to review them to investigate web related traffic. This lab also gave me a good idea about how real-world digital evidences are collected and preserved for future use to dig deeper to investigate potential case of an insider threats just by capturing their logs and network activity.

After the analysis I was able to infer that, an insider from the company is using the internal host to browse images on various external sources, with one of them being a highly malicious domain. There were also the traces of admin level logins to .evl domain leading to a malicious .jpg file trying to inject shellcode.

Glossary

1. Network Intrusion Protection Systems (NIPS) and Network Intrusion Detection Systems (NIDS) are tested on the Technologies and Tools portion of the Security+ certification exam. This article details what is covered on the Security+ certification exam regarding these important network security devices. This article should not substitute for studying but rather serve as a brief review and guide for areas that you may need to look over again.
2. Firewall Analyzer (Proxy Log Analyzer) collects and archives the proxy server logs, analyzes them, and generates useful corporate internet access information reports. Proxy server reports provide network security administrators and managed security service providers (MSSP) with important insight into the efficiency of their corporate Internet usage. As a proxy log analysis tool, Firewall Analyzer supports BlueCoat, Microsoft ISA, Squid proxy logs and servers.
3. Wireshark is the world's foremost and widely used network protocol analyzer. It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.
4. grep is a command-line utility for searching plain-text data sets for lines that match a regular expression. Its name comes from the ed command g/re/p, which has the same effect: doing a global search with the regular expression and printing all matching lines.
5. Snort is an open source network intrusion detection system (NIDS) created by Martin Roesch. ... Through protocol analysis and content searching and matching, Snort detects attack methods, including denial of service, buffer overflow, CGI attacks, stealth port scans, and SMB probes.

References

- [1] “Technologies And Tools - NIPS / NIDS.” Infosec Resources, resources.infosecinstitute.com/category/certifications-training/securityplus/sec-domains/technologies-and-tools-in-security/installing-and-configuring-network-components-to-support-organizational-security/technologies-and-tools-nips-nids/.
- [2] Hoffman, Chris. “How to Use Wireshark to Capture, Filter and Inspect Packets.” How, How-To Geek, 14 June 2017, www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/.
- [3] “Grep Command in Unix/Linux.” GeeksforGeeks, ., 20 May 2019, www.geeksforgeeks.org/grep-command-in-unixlinux/.
- [4] Rouse, Margaret. “What Is Snort? - Definition from WhatIs.com.” SearchMidmarketSecurity, TechTarget, 21 Sept. 2005, searchmidmarketsecurity.techtarget.com/definition/Snort.